

Modules

Summary

- We use modules to organize our code across multiple files.
- Objects defined in a module are private and invisible to other modules unless exported.
- We use **export** and **import** statements to export and import objects from various modules. These statements are part of the ES6 module format.
- Over years, many module formats have been developed for JavaScript. Examples are CommonJS (introduced by Node), AMD, UMD, etc.
- We can use the **module** setting in tsconfig to specify the module format the compiler should use when emitting JavaScript code.

Cheat Sheet

Exporting and importing

```
// shapes.ts
export class Circle {}
export class Square {}

// app.ts
import { Circle, Square as MySquare } from './shapes';
```

Default exports

```
// shapes.ts
export default class Circle {}

// app.ts
import Circle from './shapes';
```

Wildcard imports

```
// app.ts
import * as Shapes from './shapes';

let circle = new Shapes.Circle();
```

Re-exporting

```
// /shapes/index.ts
export { Circle } from './circle';
export { Square } from './square';

// app.ts
import { Circle, Square } from './shapes';
```